```
LL            IIIIII    BBBBBBBB    EEEEEEEEEE   DDDDDDDD      IIIIII    VV        VV
LL            IIIIII    BBBBBBBB    EEEEEEEEEE   DDDDDDDD      IIIIII    VV        VV
LL              II      BB      BB  EE           DD      DD      II      VV        VV
LL              II      BB      BB  EE           DD      DD      II      VV        VV
LL              II      BB      BB  EE           DD      DD      II      VV        VV
LL              II      BBBBBBBB    EEEEEEE      DD      DD      II      VV        VV
LL              II      BBBBBBBB    EEEEEEE      DD      DD      II      VV        VV
LL              II      BB      BB  EE           DD      DD      II      VV        VV
LL              II      BB      BB  EE           DD      DD      II       VV      VV
LL              II      BB      BB  EE           DD      DD      II        VV    VV     ....
LL              II      BB      BB  EE           DD      DD      II         VV  VV      ....
LLLLLLLLL     IIIIII    BBBBBBBB    EEEEEEEEEE   DDDDDDDD      IIIIII        VV VV      ....
LLLLLLLLL     IIIIII    BBBBBBBB    EEEEEEEEEE   DDDDDDDD      IIIIII         VV        ....

LL            IIIIII    SSSSSSSS
LL            IIIIII    SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II         SSSSSS
LL              II         SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLL     IIIIII    SSSSSSSS
LLLLLLLLL     IIIIII    SSSSSSSS
```

```
0000    1              .TITLE  LIB$EDIV - Execute EDIV instruction
0000    2              .IDENT  /1-001/                 ; File: LIBEDIV.MAR Edit: SBL1001
0000    3
0000    4      ;
0000    5      ;***********************************************************************
0000    6      ;*                                                                     *
0000    7      ;*   COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                           *
0000    8      ;*   DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.            *
0000    9      ;*   ALL RIGHTS RESERVED.                                              *
0000   10      ;*                                                                     *
0000   11      ;*   THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
0000   12      ;*   ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
0000   13      ;*   INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
0000   14      ;*   COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
0000   15      ;*   OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
0000   16      ;*   TRANSFERRED.                                                       *
0000   17      ;*                                                                     *
0000   18      ;*   THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
0000   19      ;*   AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
0000   20      ;*   CORPORATION.                                                       *
0000   21      ;*                                                                     *
0000   22      ;*   DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
0000   23      ;*   SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.            *
0000   24      ;*                                                                     *
0000   25      ;*                                                                     *
0000   26      ;***********************************************************************
0000   27      ;
0000   28
0000   29      ;++
0000   30      ; FACILITY: General Utility Library
0000   31      ;
0000   32      ; ABSTRACT:
0000   33      ;
0000   34      ;       This module contains LIB$EDIV, which makes the VAX EDIV instruction
0000   35      ;       available as a callable procedure.
0000   36      ;
0000   37      ; ENVIRONMENT: Runs at any access mode, AST Reentrant
0000   38      ;
0000   39      ; AUTHOR: Steven B. Lionel, CREATION DATE: 8-July-1981
0000   40      ;
0000   41      ; MODIFIED BY:
0000   42      ;
0000   43      ; 1-001 - Original. SBL 8-July-1981
0000   44      ;--
```

LIB$EDIV
1-001

K 2

- Execute EDIV instruction
DECLARATIONS

15-SEP-1984 23:59:10   VAX/VMS Macro V04-00   Page 2
6-SEP-1984 11:06:01   [LIBRTL.SRC]LIBEDIV.MAR;1   (2)

```
       0000    46              .SBTTL  DECLARATIONS
       0000    47      ;
       0000    48      ; LIBRARY MACRO CALLS:
       0000    49      ;
       0000    50          $SSDEF                      ; Define SS$ symbols
       0000    51          $CHFDEF                     ; Define CHF$ symbols
       0000    52      ;
       0000    53      ; EXTERNAL DECLARATIONS:
       0000    54      ;
       0000    55          .DSABL  GBL                 ; Force all external symbols to be declared
       0000    56          .EXTRN  LIB$SIG_TO_RET      ; Convert signal to return with status
       0000    57      ;
       0000    58      ; MACROS:
       0000    59      ;
       0000    60          NONE
       0000    61      ;
       0000    62      ; EQUATED SYMBOLS:
       0000    63      ;
       0000    64          NONE
       0000    65      ;
       0000    66      ; OWN STORAGE:
       0000    67      ;
       0000    68          NONE
       0000    69      ;
       0000    70      ; PSECT DECLARATIONS:
       0000    71      ;
   00000000    72          .PSECT _LIB$CODE PIC, USR, CON, REL, LCL, SHR, -
       0000    73                  EXE, RD, NOWRT, LONG
       0000    74
```

LIB$EDIV
1-001

L 2

- Execute EDIV instruction   15-SEP-1984 23:59:10 VAX/VMS Macro V04-00   Page 3
  LIB$EDIV - Execute EDIV instruction   6-SEP-1984 11:06:01 [LIBRTL.SRC]LIBEDIV.MAR;1   (3)

```
                          0000    76          .SBTTL  LIB$EDIV - Execute EDIV instruction
                          0000    77 ;++
                          0000    78 ; FUNCTIONAL DESCRIPTION:
                          0000    79 ;
                          0000    80 ;       This procedure makes the VAX EDIV instruction available as
                          0000    81 ;       a callable procedure.
                          0000    82 ;
                          0000    83 ;       The dividend argument is divided by the divisor argument;
                          0000    84 ;       the quotient argument is replaced by the quotient and the
                          0000    85 ;       remainder argument is replaced by the remainder.
                          0000    86 ;
                          0000    87 ;       For more information, see the VAX-11 Architecture Handbook.
                          0000    88 ;
                          0000    89 ; CALLING SEQUENCE:
                          0000    90 ;
                          0000    91 ;       status.wlc.v = LIB$EDIV (divisor.rl.r, dividend.rq.r,
                          0000    92 ;                                 quotient.wl.r, remainder.wl.r)
                          0000    93 ;
                          0000    94 ; FORMAL PARAMETERS:
                          0000    95 ;
                00000004  0000    96 ;       divisor = 4              ; The address of the longword integer divisor.
                          0000    97 ;
                00000008  0000    98 ;       dividend = 8             ; The address of the quadword integer dividend.
                          0000    99 ;
                0000000C  0000   100 ;       quotient = 12            ; The address of the longword integer location
                          0000   101 ;                                ; where the quotient will be stored.
                          0000   102 ;
                00000010  0000   103 ;       remainder = 16           ; The address of the longword integer location
                          0000   104 ;                                ; where the remainder will be stored.
                          0000   105 ;
                          0000   106 ; IMPLICIT INPUTS:
                          0000   107 ;
                          0000   108 ;       NONE
                          0000   109 ;
                          0000   110 ; IMPLICIT OUTPUTS:
                          0000   111 ;
                          0000   112 ;       NONE
                          0000   113 ;
                          0000   114 ; COMPLETION STATUS:
                          0000   115 ;
                          0000   116 ;       SS$_NORMAL, normal successful completion
                          0000   117 ;       SS$_INTOVF, integer overflow
                          0000   118 ;       SS$_INTDIV, integer divide by zero
                          0000   119 ;
                          0000   120 ; SIDE EFFECTS:
                          0000   121 ;
                          0000   122 ;       If integer overflow or divide-by-zero occur, then the quotient
                          0000   123 ;       operand is replaced by bits 31:0 of the dividend operand, and the
                          0000   124 ;       remainder is replaced by zero.
                          0000   125 ;
                          0000   126 ;--
                          0000   127
                4000      0000   128          .ENTRY LIB$EDIV, ^M<IV>          ; Entry point
                          0002   129                                           ; IV must be enabled
                          0002   130
 6D   13'AF   9E          0002   131          MOVAB   B^HANDLER, (FP)          ; Enable local condition handler
                          0006   132                                           ; to intercept exceptions
```

M 2

LIB$EDIV   - Execute EDIV instruction   15-SEP-1984 23:59:10 VAX/VMS Macro V04-00   Page 4
1-001   LIB$EDIV - Execute EDIV instruction   6-SEP-1984 11:06:01 [LIBRTL.SRC]LIBEDIV.MAR;1   (3)

```
10 BC   OC BC   08 BC   04 BC   7B  0006  133
                                    0006  134          EDIV    @divisor(AP), @dividend(AP), -
                                    000F  135                  @quotient(AP), @remainder(AP)
                                    000F  136
                                    000F  137  ;+
                                    000F  138  ; If the EDIV caused an exception, HANDLER will unwind to our caller
                                    000F  139  ; with the correct status.
                                    000F  140  ;-
                                    000F  141
                      50  01   DO   000F  142          MOVL    #SS$_NORMAL, R0          ; Successful completion
                               04   0012  143          RET
```

N 2

LIB$EDIV   - Execute EDIV instruction   15-SEP-1984 23:59:10 VAX/VMS Macro V04-00   Page 5
1-001   LIB$EDIV - Execute EDIV instruction   6-SEP-1984 11:06:01 [LIBRTL.SRC]LIBEDIV.MAR;1   (4)

```
                         0013   145 ;+
                         0013   146 ; Local condition handler to return $SS$_INTOVF and SS$_INTDIV as return
                         0013   147 ; statuses, all other exceptions resignal.
                         0013   148 ;-
                         0013   149
                         0013   150 HANDLER:
                  0000   0013   151          .WORD    ^M<>                          ; Entry mask
                         0015   152
         50   08 AC  DO  0015   153          MOVL     CHF$L_MCHARGLST(AP), R0   ; Get mechanism args list address
              08 A0  D5  0019   154          TSTL     CHF$L_MCH_DEPTH(R0)       ; Is signal depth zero?
                  1F  12  001C  155          BNEQ     90$                       ; If not, resignal
         50   04 AC  DO  001E  156          MOVL     CHF$L_SIGARGLST(AP), R0   ; Get signal args list address
 0000047C 8F  04 A0  D1  0022  157          CMPL     CHF$L_SIG_NAME(R0), #SS$_INTOVF
                         002A  158                                             ; Is it integer overflow?
              0A  13  002A  159          BEQL     10$                          ; If so, return with status
 00000484 8F  04 A0  D1  002C  160          CMPL     CHF$L_SIG_NAME(R0), #SS$_INTDIV
                         0034  161                                             ; Is it integer divide-by-zero?
              07  12  0034  162          BNEQ     90$                          ; If not, resignal
 00000000'GF  6C  FA  0036  163 10$:     CALLG    (AP), G^LIB$SIG_TO_RET   ; Return to our establisher's caller
                         003D  164                                             ; with condition code in R0
    50   00000918 8F  DO  003D  165 90$:     MOVL     #SS$_RESIGNAL, R0        ; Resignal (ignored if we just called
                         0044  166                                             ; LIB$SIG_TO_RET)
              04  0044  167          RET
                  0045  168
                  0045  169          .END                              ; End of module LIB$EDIV
```

```
CHF$L_MCHARGLST                 = 00000008
CHF$L_MCH_DEPTH                 = 00000008
CHF$L_SIGARGLST                 = 00000004
CHF$L_SIG_NAME                  = 00000004
DIVIDEND                        = 00000008
DIVISOR                         = 00000004
HANDLER                           00000013 R    02
LIB$EDIV                          00000000 RG   02
LIB$SIG_TO_RET                    ******** X    00
QUOTIENT                        = 0000000C
REMAINDER                       = 00000010
SS$_INTDIV                      = 00000484
SS$_INTOVF                      = 0000047C
SS$_NORMAL                      = 00000001
SS$_RESIGNAL                    = 00000918
```

```
                              +----------------+
                              ! Psect synopsis !
                              +----------------+
```

```
PSECT name                    Allocation          PSECT No.  Attributes
----------                    ----------          ---------  ----------
.  ABS  .                     00000000 (    0.)   00 (  0.)  NOPIC  USR  CON  ABS  LCL  NOSHR  NOEXE  NORD  NOWRT  NOVEC  BYTE
$ABS$                         00000000 (    0.)   01 (  1.)  NOPIC  USR  CON  ABS  LCL  NOSHR   EXE   RD    WRT   NOVEC  BYTE
_LIB$CODE                     00000045 (   69.)   02 (  2.)   PIC   USR  CON  REL  LCL   SHR    EXE   RD   NOWRT  NOVEC  LONG
```

```
                        +------------------------+
                        ! Performance indicators !
                        +------------------------+
```

```
Phase                    Page faults   CPU Time        Elapsed Time
-----                    -----------   --------        ------------
Initialization                   29    00:00:00.05     00:00:01.47
Command processing              102    00:00:00.33     00:00:03.36
Pass 1                          183    00:00:02.66     00:00:11.66
Symbol table sort                 0    00:00:00.41     00:00:01.71
Pass 2                           44    00:00:00.53     00:00:04.30
Symbol table output               4    00:00:00.02     00:00:00.02
Psect synopsis output             2    00:00:00.02     00:00:00.02
Cross-reference output            0    00:00:00.00     00:00:00.00
Assembler run totals            366    00:00:04.02     00:00:22.54
```

The working set limit was 1200 pages.
21140 bytes (42 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 427 non-local and 2 local symbols.
169 source lines were read in Pass 1, producing 13 object records in Pass 2.
9 pages of virtual memory were used to define 8 macros.

```
                        +--------------------------+
                        ! Macro library statistics !
                        +--------------------------+
```

```
Macro library name                    Macros defined
------------------                    --------------
_$255$DUA28:[SYSLIB]STARLET.MLB;2            5
```

486 GETS were required to define 5 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LIS$:LIBEDIV/OBJ=OBJ$:LIBEDIV MSRC$:LIBEDIV/UPDATE=(ENH$:LIBEDIV)

LIBEMODH
LIS

LIBEMODD
LIS

LIBESTEMU
LIS

LIBEF
LIS

LIBEMULAT
LIS

LIBFFS
LIS

LIBFINCVT
LIS

LIBE2AREV
LIS

LIBFAO
LIS

LIBEMODG
LIS

LIBEXTV
LIS

LIBEDIV
LIS

LIBESTABL
LIS

LIBFFC
LIS

LIBEMODF
LIS

LIBEMUL
LIS

LIBFILSCA
LIS

LIBEXTZV
LIS

LIBEBCASC
LIS

LIBFAOL
LIS